

\$nolist

```
codemacro$rout: PROC external;  
end$proc codemacro$rout;
```

```
db$cm$rout: PROC external;  
end$proc db$cm$rout;
```

```
dw$cm$rout: PROC external;  
end$proc dw$cm$rout;
```

```
dd$cm$rout: PROC external;  
end$proc dd$cm$rout;
```

```
segfix$cm$rout: PROC external;  
end$proc segfix$cm$rout;
```

```
nosegfix$cm$rout: PROC external;  
end$proc nosegfix$cm$rout;
```

```
modrm$cm$rout: PROC external;  
end$proc modrm$cm$rout;
```

```
relb$cm$rout: PROC external;  
end$proc relb$cm$rout;
```

```
relw$cm$rout: PROC external;  
end$proc relw$cm$rout;
```

```
dbit$cm$rout: PROC external;  
end$proc dbit$cm$rout;
```

```
end$cm$rout: PROC external;  
end$proc end$cm$rout;
```

\$list

CR/M-86 v.1.1 (Vol. IV)

COPYRIGHT - 1981, 1982

DIGITAL RESEARCH

P.O. BOX 570

BAYVIEW, MI 48060

SER. = C81-162

tnolist

/* DECLARATION FOR 'CODEMACRO' PART

Extended version of ASM36 */

dcl	cm\$error	byte	external,
	global cm\$error	byte	external,
	cm\$list\$overflow	byte	external,
	cmpt	address	external;

/* level 2 in the syntax-tree of codemacro building */

/* Procedure to initialize temporary storage and pointers
concerning the building of codemacro */

```
init$cm$rout: PROC external;  
end$proc init$cm$rout;
```

```
name$rout: PROC byte external;  
end$proc name$rout;
```

```
formal$list$rout: PROC external;  
end$proc formal$list$rout;
```

```
terminate$cm$rout: PROC external;  
end$proc terminate$cm$rout;
```

```
legal$parameter: PROC(lg,ptr,ptr2) byte external;  
dcl lg byte,  
      (ptr,ptr2) address;  
end$proc legal$parameter;
```

```
legal$seg$reg: PROC byte external;  
end$proc legal$seg$reg;
```

```
put$b: PROC(b) external;  
dcl b byte;  
end$proc put$b;
```

```
put$w: PROC(w) external;  
dcl w address;  
end$proc put$w;
```

\$list

COPYRIGHT 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

*nolist

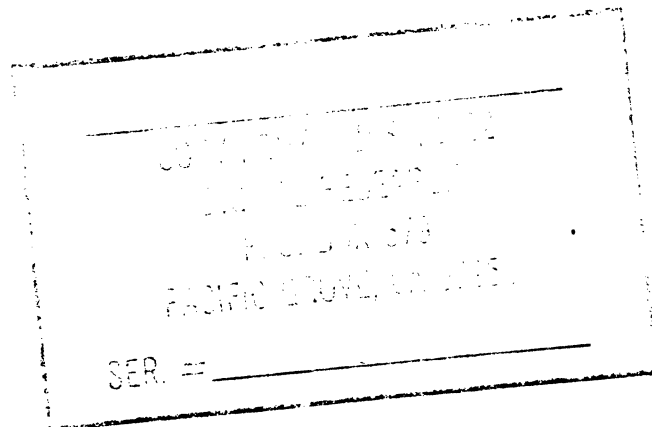
/*

modified 6/16/81 R. Silberstein

*/

declare

aaal byte external,
aad1 byte external,
aam1 byte external,
aas1 byte external,
adc1l byte external,
add1l byte external,
and10 byte external,
call3 byte external,
callf2 byte external,
cbw1 byte external,
clc1 byte external,
cld1 byte external,
cli1 byte external,
cmc1 byte external,
cmp1l byte external,
cmps2 byte external,
CMPSB1 BYTE EXTERNAL,
CMPSW1 BYTE EXTERNAL,
cwd1 byte external,
daa1 byte external,
das1 byte external,
dec3 byte external,
div2 byte external,
esc3 byte external,
hlt1 byte external,
idiv2 byte external,
imul2 byte external,
in4 byte external,
inc3 byte external,
int2 byte external,
intol byte external,
iret1 byte external,
jal byte external,
jae1 byte external,
jbl byte external,
jbe1 byte external,
jcxzl byte external,
je1 byte external,
jq1 byte external,
jge1 byte external,
jl1 byte external,
jle1 byte external,
jmp2 byte external,
jmpf2 byte external,
jmpi byte external,
jne1 byte external,
jno1 byte external,
jnp1 byte external,
jns1 byte external,



jol byte external,
jpl byte external,
jsl byte external,
lahf1 byte external,
lds1 byte external,
les1 byte external,
leal byte external,
lock1 byte external,
lods2 byte external,
LODSB1 BYTE EXTERNAL,
LODSW1 BYTE EXTERNAL,
loop1 byte external,
loopel byte external,
loopnel byte external,
mov17 byte external,
movs2 byte external,
MOVSB1 BYTE EXTERNAL,
MOVSW1 BYTE EXTERNAL,
mul2 byte external,
neg2 byte external,
nop1 byte external,
not2 byte external,
or10 byte external,
out4 byte external,
pop4 byte external,
popf1 byte external,
push3 byte external,
pushf1 byte external,
rc14 byte external,
rcr4 byte external,
rep1 byte external,
repel byte external,
repnel byte external,
ret3 byte external,
retf3 byte external,
rol4 byte external,
ror4 byte external,
sahf1 byte external,
sal4 byte external,
sar4 byte external,
sbb11 byte external,
scas2 byte external,
SCASB1 BYTE EXTERNAL,
SCASW1 BYTE EXTERNAL,
shr4 byte external,
stc1 byte external,
std1 byte external,
sti1 byte external,
stos2 byte external,
STOSB1 BYTE EXTERNAL,
STOSW1 BYTE EXTERNAL,
sub11 byte external,
test10 byte external,
wait1 byte external,
xchg6 byte external,
xlat1 byte external,
xor10 byte external;

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

#list

COPYRIGHT 1982	
DIGITAL RESEARCH	
A.O. BOX 27	
PACIFIC GROVE, CA 93950	
SER. =	_____

\$nolist

clearcmindex: proc external;
end clearcmindex;

emit: proc external; /* emit codebytes for an instruction */
end emit;

emitdummies: proc external; /* emit dummy (NO-OPs) bytes if error */
end emitdummies;

commandtype: proc(commo,lg,pt) byte external;
dcl (commo,lg) byte,pt address;
end commandtype;

mDBNROUT: proc external;
end mDBNROUT;

mDBFROUT: proc external;
end mDBFROUT;

mDWNROUT: proc external;
end mDWNROUT;

mDWFROUT: proc external;
end mDWFROUT;

mDDFROUT: proc external;
end mDDFROUT;

mRELBROUT: proc external;
end mRELBROUT;

mRELWROUT: proc external;
end mRELWROUT;

mNOSEGFIXROUT: proc external;
end mNOSEGFIXROUT;

mSEGFIXROUT: proc external;
end mSEGFIXROUT;

mMODRM1ROUT: proc external;
end mMODRM1ROUT;

mMODRM2ROUT: proc external;
end mMODRM2ROUT;

mDBITROUT: proc external;
end mDBITROUT;

/* test if operands match instruction */
searchformatch: proc byte external;
end searchformatch;

\$list

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

DLINE.EXT;4

25-MAR-1982 20:15:28.68

Page 1

#nolist

decodeline: proc external;
end decodeline;

#list

DATE: 25-MAR-1982
TIME: 20:15:28.68
PAGE: 1
FILE: DLINE.EXT
SERIAL: _____

%nolist

```
errmsg: proc(errno) external;  
  dcl errno byte;  
end errmsg;
```

%list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

#nolist

```
dcl
cip      addr external, /* current instruction pointer */
csegtype byte external, /* current segment type, code,data */
csegvalue addr external, /* current segment value */
csegspec byte external, /* true if segment value specified */
dspec    byte external,
curdseg   addr external, /* current data segment value */
token     struc(        /* actual token scanned */
    type  byte,
    descr byte,
    value addr) external, /* token value */
nextch    byte external, /* next input character */
acclen     byte external, /* accumulator length */
accum(80)  byte external, /* actual token scanned */
nooper     byte external, /* no of instruction operands */
operands(4) operandstruc /* instruction operands,max 4 */
            external;
```

#list

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

#nolist

/*

modified 8/19/81 R. Silberstein

*/

```
operand: proc byte external;  
end operand;
```

```
NOFORWARDOPER: PROC BYTE EXTERNAL;  
END NOFORWARDOPER;
```

```
expression: proc(pt) byte external;  
    dcl pt address;  
end expression;
```

```
noforwardexpr: proc(pt) byte external;  
    dcl pt address;  
end noforwardexpr;
```

#list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

/*

modified 3/28/81 R. Silberstein

modified 3/16/81 R. Silberstein

*/

outhexbyte: proc(ch) external;
 dcl ch byte;
end outhexbyte;

outprintbyte: proc(ch) external;
 dcl ch byte;
end outprintbyte;

outsymbolbyte: proc(ch) external;
 dcl ch byte;
end outsymbolbyte;

insourcebyte: proc byte external;
end insourcebyte;

inincludebyte: proc byte external;
end inincludebyte;

opensource: proc external;
end opensource;

openinclude: proc external;
end openinclude;

openhex: proc external;
end openhex;

openprint: proc external;
end openprint;

opensymbol: proc external;
end opensymbol;

close\$source: proc external;
end close\$source;

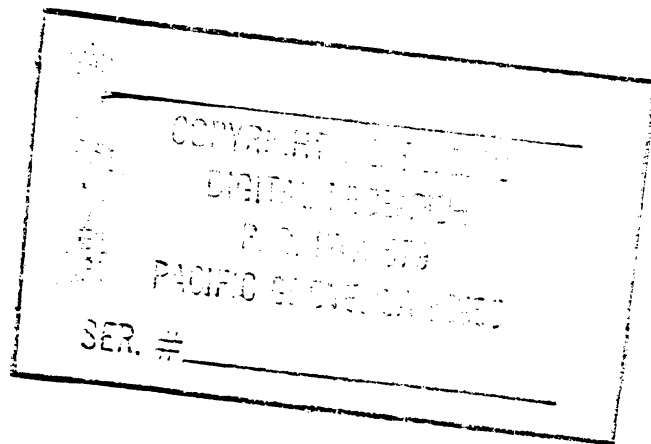
rewindsource: proc external;
end rewindsource;

close\$include: proc external;
end close\$include;

closehex: proc external;
end closehex;

closeprint: proc external;
end closeprint;

closesymbol: proc external;



end closesymbol;

i\$file\$setup: proc(dev,filnam,filtyp) external;
 dcl dev byte,(filnam,filtyp) addr;
end i\$file\$setup;

filesetup: proc byte external;
end filesetup;

*list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

Enolist

11

modified 3/28/81 R. Silberstein
modified 4/16/81 R. Silberstein
modified 7/24/81 R. Silberstein
modified 7/2/81 R. Silberstein

新

del

```
pass      byte external, /* current pass no, 1,2,3 */
```

```
/* address counters */
```

cip	addr external,	/* current instruction pointer */
csegtype	byte external,	/* current segment type, code,data, stack or extra data */
csegvalue	addr external,	/* current segment value */
csegspec	byte external,	/* true if segment value specified */
escip	addr external,	/* current ES instruction pointer */
cscip	addr external,	/* current CS instruction pointer */
sscip	addr external,	/* current SS instruction pointer */
dscip	addr external,	/* current DS instruction pointer */
curcseg	addr external,	/* current code segment value */
curdseg	addr external,	/* current data segment value */
curseg	addr external,	/* current stack segment value */
curseg	addr external,	/* current extra segment value */
cspec	byte external,	/* true if code segm. value given */
dspec	byte external,	/* true if data segm. value given */
sspec	byte external,	/* true if stack segment given */
espec	byte external,	/* true if extra segment given */

```
/* print output parameters */
```

```

printon          byte external, /* on/off flag */
printswitchoff   byte external, /* set/reset by NOLIST/LIST */
IFLIST          BYTE EXTERNAL, /* SET/RESET BY IFLIST/NOIFLIST */
maxcol          byte external, /* pagewidth */
sourcename (12) byte external, /* source file name */
savesource (12) byte external, /* source file during INCLUDE file */
printdevice      byte external, /* printfile device */
SYMBOLDEVICE     BYTE EXTERNAL, /* SYMBOL FILE DEVICE */
title (30)       byte external, /* user specified program title */
pagesize         byte external, /* page size */
simform          byte external, /* true if formfeed simulation */
sourcebuf (30)   byte external, /* source input to be printed */
sourceptr        byte external, /* source buffer pointer */
prefix (240)     byte external, /* prefix to source line */
prefixptr        byte external, /* pointer to prefix buffer */
ABSADDR (4)      BYTE EXTERNAL, /* ABSOLUTE ADDRESS FIELD */

```

```
/* io error status */
```

```
errors      addr external,      /* counts no of errors */
```

```
/* scanner variables: */
```

```
token  struc(          /* actual token scanned */
        type  byte,      /* token type, legal values :
                        reg      - register
                        pseudo   - pseudo code
                        string   - text string
                        spec     - special character
                        number   - number
                        operator - arithmetic operator
                        ident    - identifier */
```

```
        descr  byte,      /* token description, legal values:
                        nil      - no specification
                        byte     - 8 bit type
                        word     - 16 bit type
                        dword    - 32 bit type */
```

```
        value  addr) external, /* token value */
nextch      byte external, /* next input character */
acclen      byte external, /* accumulator length */
accum(80)   byte external, /* actual token scanned */
accumsave(80) byte external,
acclensave  byte external,
eofset      byte external, /* true if end-of-file found */
```

```
/* Miscellaneous variables: */
```

```
intelhex$on  byte external, /* true if INTEL hex format */
noerror      byte external, /* codemacro decoding errorflag */
errorprinted byte external, /* true if an error is printed */
firstmacroptr address external, /* pointer at first codemacro */
macroptr     address external, /* current pointer within macros */
fullsymtab   byte external, /* full if symboltable is full */
include$on   byte external, /* true if INCLUDE file input */
IFLEVEL      BYTE EXTERNAL, /* IF-ENDIF NESTING LEVEL */
currentsymbol symbolstruc /* current scanned symbol */
              external,
symtabaddr   address external, /* pointer at symbol in table */
nooper       byte external, /* no of instruction operands */
operands(4)  operandstruc /* instruction operands, max 4 */
              external,
codemacroptr address external, /* pointer to found codemacro */
help(5)      byte external, /* ascii number scratch area */
i            byte external, /* scratch variable */
default$drive byte external, /* default disk drive */
include$default byte external, /* default drive for include file */
codemacro$flag byte external, /* true if building a codemacro */
```

```
globalinit: procedure external; /* initiate some globals */
end globalinit;
```

```
$list
```

\$nolist

instruction: proc external;
end instruction;

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$nolist
```

```
/* Template for all BDOS calls */
```

```
/*
```

```
modified 3/26/81 R. Silberstein
```

```
modified 7/14/81 R. Silberstein
```

```
*/
```

```
mon1: procedure(func,info) external;
```

```
  declare func byte,  
          info address;
```

```
end mon1;
```

```
mon2: procedure(func,info) byte external;
```

```
  declare func byte,  
          info address;
```

```
end mon2;
```

```
declare fcb(1) byte external;
```

```
declare fcb16(1) byte external;
```

```
declare tbuff(1) byte external;
```

```
declare endbuf address external;
```

```

/*****
*
*      B D O S   Externals
*
*****/
```

```
system$reset:
```

```
  procedure external;  
  end system$reset;
```

```
read$console:
```

```
  procedure byte external;  
  end read$console;
```

```
write$console:
```

```
  procedure (char) external;  
  declare char byte;  
  end write$console;
```

```
write$list:
```

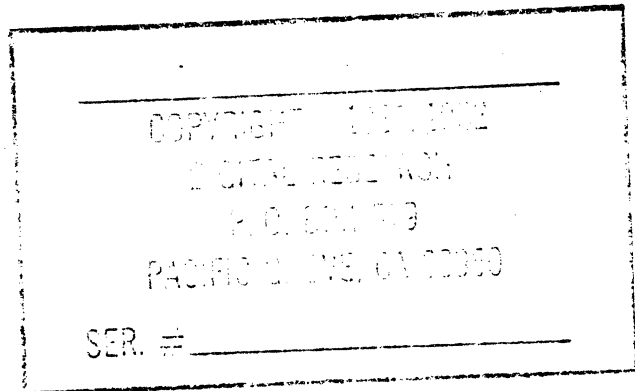
```
  procedure (char) external;  
  declare char byte;  
  end write$list;
```

```
constat:
```

```
  procedure byte external;  
  end constat;
```

```
VERSION: PROCEDURE ADDRESS EXTERNAL;
```

```
END VERSION;
```




```
select$disk:
  procedure (disk$number) external;
  declare disk$number byte;
  end select$disk;

open$file:
  procedure (fcb$address) byte external;
  declare fcb$address address;
  end open$file;

OPEN$RO$FILE: PROCEDURE (FCB$ADDRESS) BYTE EXTERNAL;
  DECLARE FCB$ADDRESS ADDRESS;
  END OPEN$RO$FILE;

close$file:
  procedure (fcb$address) byte external;
  declare fcb$address address;
  end close$file;

delete$file:
  procedure (fcb$address) external;
  declare fcb$address address;
  end delete$file;

read$record:
  procedure (fcb$address) byte external;
  declare fcb$address address;
  end read$record;

write$record:
  procedure (fcb$address) byte external;
  declare fcb$address address;
  end write$record;

create$file:
  procedure (fcb$address) byte external;
  declare fcb$address address;
  end create$file;

interrogate$disk:
  procedure byte external;
  end interrogate$disk;

set$DMA$address:
  procedure (DMA$address) external;
  declare DMA$address address;
  end set$DMA$address;

crlf: procedure external;
  end crlf;
```

```
$list
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$nolist
```

```
/*
```

```
modified 3/28/81 R. Silberstein
```

```
*/
```

```
dcl
```

```
pass          byte external, /* current pass no, 1,2,3 */
```

```
/* address counters */
```

```
cip           addr external, /* current instruction pointer */  
csegtype      byte external, /* current segment type, code,data,  
                        stack or extra data */
```

```
/* print output parameters */
```

```
print$on      byte external, /* on/off flag */  
printswitchoff byte external, /* set/reset by NOLIST/LIST */  
IFLIST        BYTE EXTERNAL, /* SET/RESET BY IFLIST/NOIFLIST */  
maxcol        byte external, /* pagewidth */  
printdevice   byte external, /* printfile device */  
simform       byte external, /* true if formfeed simulation */
```

```
/* io error status */
```

```
errors        addr external, /* counts no of errors */  
eofset        byte external, /* true if end-of-file found */
```

```
/* Miscellaneous variables: */
```

```
errorprinted  byte external, /* true if an error is printed */  
fullsyntab    byte external, /* full if symboltable is full */  
include$on    byte external, /* true if INCLUDEfile input */  
IFLEVEL       BYTE EXTERNAL, /* IF-ENDIF NESTING LEVEL */  
help(5)       byte external, /* ascii number scratch area */  
i             byte external; /* scratch variable */
```

```
globalinit: procedure external; /* initiate some globals */  
end globalinit;
```

```
$list
```

COPYRIGHT - 1991, 1982

DENTAL RESEARCH

P. O. BOX 879

PACIFIC GROVE, CA 93950

SER. =

*nolist

/*

modified 6/16/81 R. Silberstein

*/

dcl
push byte external,
repz byte external,
aaa byte external,
movs byte external,
pushf byte external,
MOVSB BYTE EXTERNAL,
adc byte external,
add byte external,
CMPSB BYTE EXTERNAL,
ja byte external,
dec byte external,
loopne byte external,
repnz byte external,
jae byte external,
jg byte external,
clc byte external,
iand byte external,
loopz byte external,
aas byte external,
jl byte external,
in byte external,
cli byte external,
jo byte external,
inc byte external,
lahf byte external,
icall byte external,
jne byte external,
cwd byte external,
jnbe byte external,
cmp byte external,
ior byte external,
callf byte external,
div byte external,
les byte external,
sar byte external,
jmp byte external,
hlt byte external,
lock byte external,
xchg byte external,
ret byte external,
idiv byte external,
jmpf byte external,
mul byte external,
pop byte external,
sti byte external,
inot byte external,
mov byte external,
cmps byte external,
iret byte external,

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

popf byte external,
imul byte external,
out byte external,
xlat byte external,
japs byte external,
loope byte external;

\$list

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE CA 93950

SER. #

\$nolist

emitinit: proc external;
end emitinit;

emitterminate: proc external;
end emitterminate;

emitcodebyte: proc(ch,typ) external;
 dcl (ch,typ) byte;
end emitcodebyte;

\$list

CP/M-86 v.1.1 (Vol. IV)

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # C81-162

```
$nolist
pfind: proc (n,s,a) byte external;
  dcl
    n      byte,
    (s,a)  address;
end pfind;
$list
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER #:

```
$nolist
```

```
/*
```

```
    modified 7/27/81 R. Silberstein
```

```
*/
```

```
printsinglebyte: proc(ch) external;  
    dcl ch byte;  
end printsinglebyte;
```

```
printCrLf: proc external;  
end printCrLf;
```

```
printnewpage: proc external;  
end printnewpage;
```

```
printsourceline: proc external;  
end printsourceline;
```

```
printinit: proc external;  
end printinit;
```

```
printterminate: proc (USEFACT) external;  
    DECLARE USEFACT BYTE;  
end printterminate;
```

```
eject: proc external;  
end eject;
```

```
$list
```

COPYRIGHT 1981, 1982
DIGITAL RESEARCH
P.O. BOX 379
PACIFIC GROVE, CA 93950

VER. # _____

```
$nolist
```

```
/*
```

```
    modified 4/9/81  R. Silberstein
```

```
*/
```

```
LISTCIP: PROCEDURE EXTERNAL;  
END LISTCIP;
```

```
DBrout: proc external;  
end DBrout;
```

```
DWrout: proc external;  
end DWrout;
```

```
DDrout: proc external;  
end DDrout;
```

```
RSrout: proc (typ) external;  
    dcl typ byte;  
end RSrout;
```

```
CSEGrout: proc external;  
end CSEGrout;
```

```
DSEGrout: proc external;  
end DSEGrout;
```

```
SSEGrout: proc external;  
end SSEGrout;
```

```
ESEGrout: proc external;  
end ESEGrout;
```

```
ORGrout: proc external;  
end ORGrout;
```

```
EQUrout: proc external;  
end EQUrout;
```

```
$list
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

#nolist

/*

modified 7/24/81 R. Silberstein

*/

IFrout: proc external;
end IFrout;ENDIFrout: proc external;
end ENDIFrout;INCLUDErout: proc external;
end INCLUDErout;ENDrout: proc external;
end ENDrout;PAGESIZERout: proc external;
end PAGESIZERout;PAGEWIDTHrout: proc external;
end PAGEWIDTHrout;TITLERout: proc external;
end TITLERout;EJECTrout: proc external;
end EJECTrout;SIMFORMrout: proc external;
end SIMFORMrout;LISTrout: proc external;
end LISTrout;NOLISTRout: proc external;
end NOLISTRout;IFLISTROUT: PROC EXTERNAL;
END IFLISTROUT;NOIFLISTROUT: PROC EXTERNAL;
END NOIFLISTROUT;

#list

COPYRIGHT © 1981, 1982
E. J. RESEARCH
P. O. BOX 573
PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

scaninit: proc external;
end scaninit;

scan: proc external;
end scan;

skip\$rest\$of\$line: proc external;
end skip\$rest\$of\$line;

specialtoken: proc (tok) byte external;
 dcl tok byte;
end specialtoken;

skip\$until: proc (tok) byte external;
 dcl tok byte;
end skip\$until;

emptyline: proc byte external;
end emptyline;

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

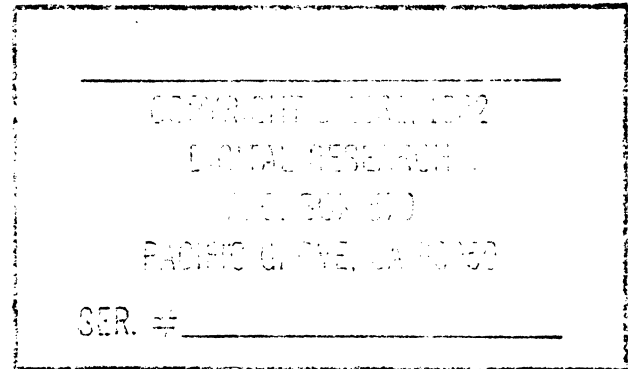
*nolist

```
typecalc: proc(val) byte external;  
  dcl val addr;  
end typecalc;
```

```
wrdtest: proc(n) byte external;  
  dcl n addr;  
end wrdtest;
```

```
copy: procedure(n,s,d) external;  
  dcl n byte,  
  (s,d) addr;  
end copy;
```

*list



inolist

/t

modified 3/26/81 R. Silberstein

modified 3/28/81 R. Silberstein

t/

outtext: procedure (t) external;

dcl t addr;

end outtext;

FILEABORT: PROCEDURE (FCBADR, TEXTADR) EXTERNAL;

DECLARE (FCBADR, TEXTADR) ADDRESS;

END FILEABORT;

fill: procedure(ch,n,pt) external;

dcl (ch,n) byte,pt addr;

end fill;

digit: procedure(ch) byte external;

dcl ch byte;

end digit;

letter: procedure(ch) byte external;

dcl ch byte;

end letter;

alphanumeric: proc(ch) byte external;

dcl ch byte;

end alphanumeric;

asciichar: proc(ch) byte external;

dcl ch byte;

end asciichar;

upper: procedure(ch) byte external;

dcl ch byte;

end upper;

equal: procedure(n,s,d) byte external;

dcl n byte,

(s,d) addr;

end equal;

hex1out: proc(n,d) external;

dcl n byte,d addr;

end hex1out;

hex2out: proc(n,d) external;

dcl n addr,d addr;

end hex2out;

decout: proc(n,d) external;

dcl n addr,d addr;

end decout;

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$nolist
```

```
/*
```

```
    modified 7/24/81 R. Silberstein
```

```
*/
```

```
dcl freeptr addr external;  
DCL END$OF$SYMBTAB ADDR EXTERNAL;
```

```
symbterminate: proc external;  
end symbterminate;
```

```
symbinit: proc external;  
end symbinit;
```

```
newsymbol: proc(lg, stradr, result) byte external;  
    dcl lg byte, (stradr, result) addr;  
end newsymbol;
```

```
newmacro: proc(lg, stradr, macdefptr) byte external;  
    dcl lg byte, (stradr, macdefptr) addr;  
end newmacro;
```

```
findsymbol: proc(lg, stradr, result) byte external;  
    dcl lg byte, (stradr, result) addr;  
end findsymbol;
```

```
getattributes: proc(symbadr, dest) external;  
    dcl (symbadr, dest) addr;  
end getattributes;
```

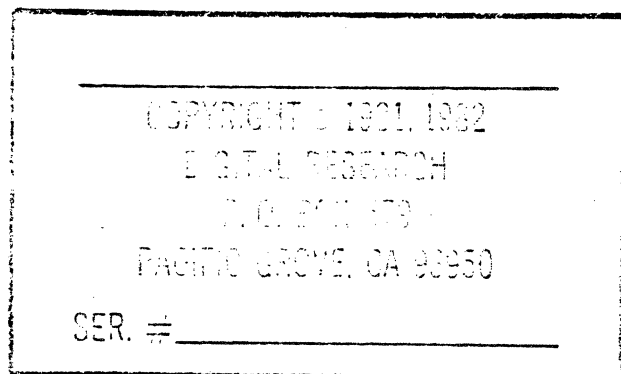
```
enterattributes: proc(symbadr, source) external;  
    dcl (symbadr, source) addr;  
end enterattributes;
```

```
findcodemacro: proc(lg, stradr, result) byte external;  
    dcl lg byte, (stradr, result) addr;  
end findcodemacro;
```

```
new$cm$body: PROC(lg, ptr) byte external;  
    dcl lg byte, ptr addr;  
end$proc new$cm$body;
```

```
new$cm$name: PROC(lg, asciiptr, returnptr) byte external;  
    dcl lg byte, (asciiptr, returnptr) addr;  
end$proc new$cm$name;
```

```
$list
```



\$nolist

/*

modified 3/28/81 R. Silberstein
modified 3/30/81 R. Silberstein
modified 7/24/81 R. Silberstein

*/

/* Text strings: */

dcl

yes byte external,
no byte external,
initials(1) byte external,
pagetext(1) byte external,
endtext(1) byte external,
USEFACTOR (1) BYTE EXTERNAL,
asm86text(1) byte external,
parerrtext(1) byte external,
openerrtext(1) byte external,
DISKREADERRTEXT (1) BYTE EXTERNAL,
DISKWRITEERRTEXT (1) BYTE EXTERNAL,
MAKEERRTEXT (1) BYTE EXTERNAL,
CLOSEERRTEXT (1) BYTE EXTERNAL,
SYMBFULLERRTEXT (1) BYTE EXTERNAL,
usbreaktext(1) byte external,
pass0text (1) byte external,
pass1text (1) byte external;

\$list

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

